A

Project Report On

HUMAN INTRUSION DETECTION SYSTEM

submitted as partial fulfilment for the award of

BACHELOR OF TECHNOLOGY

SESSION 2024-25

In

COMPUTER SCIENCE

BY

Archit Kaushik - 2100290120046

Asmit Tyagi- 2100290120058

Arun Kumar - 2000290120042

Ashish Sikarwar - 2100290120053

Under the Supervision of

Mr. Sreesh Gaur

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University

Lucknow

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Signature:

Name: Archit Kaushik

Roll No.:2100290120046

Signature:

Name: Arun Kumar

Roll No.: 2000290120042

Signature:

Name: Asmit Tyagi

Roll No.: 2100290120058

Signature:

Name: Ashish Sikarwar

Roll No.: 2100290120053

CERTIFICATE

This is to certify that Project Report entitled “HUMAN INTRUSION DETECTION SYSTEM", PCS25-45 which is submitted by Mr. Archit Kaushik, Mr. Asmit Tyagi, Mr. Arun Kumar and Mr. Ashish Sikarwar of VIII semester for Project in partial fulfilment of the requirement for the award of degree B.Tech in Department of Electronics & Communication Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Date:

11-02-2025

Supervisor Name:

Mr. Sreesh Gaur

(Signature)

ACKNOWLEDGEMENT

We are very pleased to present you with the report of the B. Tech Final Year project. We owe a special thanks to the Supervisor, Department of Electronics and Communication Engineering, KIET, Ghaziabad, for his ongoing support and guidance throughout our work. We have been constantly inspired by his sincerity, rigor, and perseverance. It was only his conscious efforts that got off the ground. We also take the opportunity to acknowledge the contribution of Mr. Sreesh Gaur, Assistant Professor of Computer Science, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project. We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/industry person/any person, of the department for their kind assistance and cooperation during the development of our project. Finally, we welcome the contribution our friends have made to the project.

Signature:

Name: Archit Kaushik
Roll No.:2100290120046

Signature:

Name: Arun Kumar
Roll No.: 2000290120042

Signature:

Name: Asmit Tyagi
Roll No.: 2100290120058

Signature:

Name: Ashish Sikarwar
Roll No.: 2100290120053

ABSTRACT

Defence installations are critical assets that require robust and sophisticated security measures to safeguard against unauthorized access, potential threats, and breaches that could compromise national security. Traditional intrusion detection systems (IDS) often face significant challenges in adapting to dynamic and complex environments typically associated with defence settings. These conventional systems struggle to effectively distinguish between genuine threats and false alarms, leading to inefficiencies and potential vulnerabilities. The need for a more adaptive, accurate, and efficient security solution has become increasingly vital to address the evolving landscape of security threats.

Human Intrusion Detection Systems play a crucial role in modern security applications, ensuring real-time monitoring and threat prevention. This project presents a deep learning-based intrusion detection system leveraging a sequential neural network. The model architecture combines a TimeDistributed Convolutional Neural Network (CNN) for spatial feature extraction, a Long Short-Term Memory (LSTM) network for capturing temporal dependencies, and fully connected layers for classification. The system processes video input sequences of 160 frames per time step, allowing for robust anomaly detection in real-time. Performance evaluation using standard metrics demonstrates high accuracy and efficiency, making the proposed system a reliable solution for intelligent surveillance.

The system is meticulously designed to operate efficiently under diverse environmental conditions, including varying lighting, weather, and terrain, which are common in defence settings. Its architecture allows seamless integration with existing defence infrastructure, such as surveillance cameras, motion detectors, and sensor networks, facilitating timely detection and rapid response to potential security breaches. Through rigorous testing and evaluation, the proposed HIDS demonstrates superior accuracy, adaptability, and operational efficiency compared to traditional systems. By bolstering the security framework of defence installations, this advanced intrusion detection system significantly contributes to enhancing national security and mitigating the risks posed by unauthorized intrusions.

LIST OF FIGURES

FIGURES	DESCRIPTION	PAGE NO.
Figure-1	Flow Chart	7
Figure-2	Sequence Diagram	19
Figure-3	Timeline Diagram	20
Figure-4	Input and Output	32
Figure-5	Anomaly(Shoplifting)	33
Figure-6	Normal	34
Figure-7	Anomaly(Fighting)	34

LIST OF TABLES

TABLE	DESCRIPTION	PAGE NO.
1	BVA	37
2	Equivalence Class	37
3	Stress Testing	38

LIST OF ABBREVATIONS

- HIDS: Human Intrusion Detection System
- CNN: Convolutional Neural Network
- LSTM: Long Short-Term Memory
- GPU: Graphics Processing Unit
- FPS: Frames Per Second
- AUC: Area Under the Curve

SDG’s MAPPING WITH JUSTIFICATION

SDG 9: Industry, Innovation, and Infrastructure

- The project leverages deep learning and artificial intelligence to enhance surveillance and security infrastructure.
- Encourages technological advancements in automated security systems for smart cities and industries.

SDG 11: Sustainable Cities and Communities

- Helps in preventing unauthorized access to restricted or sensitive areas.
- Contributes to smart surveillance systems in urban planning for safer public spaces.

SDG 16: Peace, Justice, and Strong Institutions

- Helps in real-time detection of suspicious activities, reducing threats like theft, vandalism, and assaults.
- Supports law enforcement and security agencies with advanced monitoring tools.

TABLE OF CONTENTS

DECLARATION ..... b

CERTIFICATE c

ACKNOWLEDGEMENTS ..... d

ABSTRACT ..... e

LIST OF FIGURES ..... f

LIST OF TABLES ..... g

LIST OF ABBREVATIONS ..... h

SDG's MAPPING WITH JUSTIFICATION i

CHAPTER 1 (INTRODUCTION) 1

1.1 Project Category 1

1.2 Objectives 2

1.3 Structure of Report 3

1.4 Literature Survey 4

1.5 Methodology Survey 6

CHAPTER 2 (RESEARCH WORK) 9

2.1 Abstract 9

2.2. Working Process 12

CHAPTER 3 (REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION) 15

3.1. Feasibility Study 15

3.2. Software Requirement Specification 16

3.3. SDLC Model Used.....18

CHAPTER 4 (IMPLEMNTATION) 19

4.1. Hardware Resources 20

4.2. Software Resources 23

4.3. Dataset 25

4.4. Human Resources 26

4.5. Other Resources 28

4.6. Workflow 29

CHAPTER 5 (RESULTS) 32

5.1. Performance Metrices 32

5.2. Visualization of Result 33

5.3. Test Proofs and Validation 34

5.4. System Output and Functionality 36

CHAPTER 6 (CONCLUSIONS) 39

CHAPTER 7 (FUTURE SCOPE) 41

7.1. Potential Improvements 41

7.2. Research Opportunities 41

7.3. Long term Vision 42

CHAPTER 8 (References) 43

CHAPTER 1 (INTRODUCTION)

Security and surveillance are critical concerns in modern society, requiring advanced technologies to enhance real-time monitoring and threat detection. Traditional surveillance systems rely on manual monitoring, which is prone to inefficiencies and human error. With the advancement of deep learning, intelligent intrusion detection systems have emerged as a robust alternative, offering automated and highly accurate threat identification.

This project presents a Human Intrusion Detection System utilizing a Sequential Neural Network that processes video sequences to detect unauthorized access. The system incorporates TimeDistributed Convolutional Neural Networks (CNNs) for spatial feature extraction, Long Short-Term Memory (LSTM) networks for temporal analysis, and a classification layer for intrusion detection. By analyzing 160 frames per sequence, the model ensures effective recognition of suspicious activities in real-time.

The proposed system is designed to enhance security in various applications, including surveillance in restricted areas, anomaly detection in smart homes, and public safety monitoring. By leveraging deep learning, the system provides high accuracy, reduced false alarms, and real-time processing capabilities.

1.1 Project Category

1. This project falls under the category of Artificial Intelligence (AI) and Deep Learning-based Security Systems. AI has revolutionized surveillance and security applications by introducing automation, real-time threat detection, and intelligent monitoring. The Human Intrusion Detection System (HIDS) leverages computer vision, neural networks, and deep learning to analyze video data, identify unauthorized access, and generate real-time alerts.
2. This system is specifically designed for applications in smart surveillance, security automation, and anomaly detection. Traditional security measures, such as closed-circuit television (CCTV) monitoring, rely heavily on human operators, leading to potential delays, oversights, and inefficiencies. By integrating AI, the proposed system enhances real-time processing, scalability, and adaptability, making it suitable for environments such as military bases, corporate offices, airports, and residential complexes.
3. HIDS employs a sequential neural network architecture, combining TimeDistributed Convolutional Neural Networks (CNNs) for spatial feature extraction and Long Short-

Term Memory (LSTM) networks for temporal pattern recognition. The fusion of these techniques enables accurate detection of suspicious movements and human intrusions with minimal false positives.

By utilizing advanced deep learning frameworks, edge computing, and real-time alert systems, this project aligns with the growing need for autonomous security solutions that can function with high efficiency in various environments, including low-light conditions, crowded spaces, and complex terrains.

1.2 Objectives

The primary objectives of this project are as follows:

1. Developing an AI-based Human Intrusion Detection System
2. The core goal is to create an automated security solution capable of detecting human intrusions in real-time using advanced deep learning techniques.
3. The system will replace traditional manual surveillance with automated alerts, improving security response times.
4. Implementing TimeDistributed CNN for Spatial Feature Extraction
5. By leveraging CNN-based spatial feature extraction, the system will accurately detect human figures and movements.
6. Feature extraction ensures that the system can differentiate between humans, animals, and non-intrusive objects, reducing false alarms.
7. Integrating LSTM for Temporal Pattern Recognition
8. Traditional detection systems often fail to capture motion patterns effectively.
9. LSTM enables sequence learning, allowing the system to detect suspicious behavior patterns over time, such as loitering or unauthorized movement.
10. Optimizing the Model for Real-Time Processing
11. Real-time performance is crucial for practical implementation.
12. The project focuses on reducing processing latency by optimizing model architecture and implementing techniques such as model pruning and quantization to enhance efficiency.
13. Enhancing Security Measures through Automation
14. The HIDS system aims to reduce reliance on manual monitoring, which is prone to human error and fatigue.

15. The AI-driven system will provide instant notifications and alerts, allowing security personnel to respond swiftly.
16. Ensuring Scalability and Adaptability
17. The system is designed to work in various environments, including indoor, outdoor, high-security zones, and public spaces.
18. It will support integration with existing security infrastructure, such as CCTV networks, IoT devices, and cloud-based surveillance platforms.

By achieving these objectives, this project contributes to the advancement of AI-powered security systems, making them more efficient, accurate, and scalable for real-world applications.

1.3 Structure of the Report

1. This report is structured to provide a comprehensive analysis of the development, implementation, and evaluation of the Human Intrusion Detection System. Each chapter covers specific aspects of the project, ensuring a logical flow of information and thorough documentation of the research and development process.
2. Introduction
3. Provides background information on security challenges, traditional surveillance limitations, and the need for AI-powered intrusion detection.
4. Outlines the project category, objectives, and scope of the report.
5. Literature Review
6. Examines previous research in the field of intrusion detection, deep learning-based security systems, and real-time surveillance techniques.
7. Discusses existing CNN, RNN, and hybrid AI models and their performance in intrusion detection applications.
8. Identifies gaps in current research and highlights the novel contributions of this project.
9. Methodology
10. Describes the step-by-step approach to system development, including dataset selection, preprocessing, model architecture, training, and optimization.
11. Provides technical details on the CNN-LSTM architecture and justifies its effectiveness in detecting human intrusions.
12. Discusses hardware and software requirements for implementation.
13. Analysis and Discussion

14. Explores real-world applications, potential security threats, and the effectiveness of AI-driven surveillance.
15. Includes case studies demonstrating the system's performance in different environments.
16. Analyzes challenges, limitations, and potential improvements in the detection model.
17. Results
18. Presents detailed performance metrics, including accuracy, precision, recall, F1-score, and processing speed.
19. Visualizes results through graphs, confusion matrices, and comparative analysis with other models.
20. Includes real-world test cases and their outcomes.
21. Conclusion
22. Summarizes the key findings and contributions of the project.
23. Discusses limitations and challenges, such as handling occlusions, varying lighting conditions, and scalability.
24. Provides recommendations for future enhancements, including multi-modal sensor fusion and edge-based real-time processing.
25. Future Scope
26. Identifies potential areas for improvement, such as behavioral analysis, AI explainability, and integration with advanced security infrastructure.
27. Explores research opportunities in anomaly detection, adversarial robustness, and self-learning AI models.
28. References
29. Provides a detailed bibliography of research papers, datasets, books, and online resources consulted in the study.

By structuring the report in this manner, it ensures a systematic approach to explaining the concepts, methodologies, findings, and implications of the Human Intrusion Detection System.

1.4 Literature survey

The field of Human Intrusion Detection Systems (HIDS) has evolved significantly with advancements in deep learning and computer vision technologies. This literature survey reviews key studies related to the technologies and problem statement of this project, focusing

on the application of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for real-time human intrusion detection in surveillance systems.

1. Smart Surveillance as an Edge Network Service: from Haar-Cascade, SVM to a Lightweight CNN (Nikouei et al., 2018)
   This study explores the feasibility of human-object detection schemes at the edge and introduces a lightweight Convolutional Neural Network (L-CNN) for human detection. The algorithms are validated using real-world campus surveillance video streams and open datasets, demonstrating efficient performance in resource-constrained environments.
2. Video Anomaly Detection System Using Deep Convolutional and Recurrent Models (Qasim & Verdú, 2023)
   This research presents a system that utilizes deep convolutional and recurrent models to detect anomalies in video surveillance, focusing on identifying unusual activities that may indicate security threats. The hybrid approach effectively captures both spatial and temporal features, enhancing intrusion detection accuracy.
3. Real-World Anomaly Detection in Surveillance Videos (Sultani et al., 2018)
   This paper introduces a method to learn anomalies by exploiting both normal and anomalous videos through a deep multiple instance ranking framework. The study presents a large-scale dataset of real-world surveillance videos encompassing various anomalies such as fighting, road accidents, burglary, and robbery, showcasing strong performance in real-world scenarios.
4. Spatiotemporal Analysis Using Recurrent Neural Networks for Intrusion Detection (Alzubi et al., 2022)
   This research focuses on using RNNs to analyze temporal sequences in surveillance data. The approach helps detect continuous and evolving patterns of intrusion, making it suitable for dynamic and real-world security environments.
5. Review of Human Intrusion Detection Systems Based on Deep Learning (Khani & Kazemi, 2023)
   Khani and Kazemi provide a comprehensive review of deep learning-based HIDS, discussing the advantages and limitations of different models, including CNNs and RNNs. The paper highlights key challenges and potential research directions for improving HIDS technologies.

1.5 Methodology survey

The methodology of this project focuses on developing a deep learning-based Human Intrusion Detection System using a Sequential Neural Network. The approach integrates both spatial and temporal feature extraction techniques to detect unauthorized access with high accuracy.

1. Data Preprocessing

   Frame Extraction: The input video is divided into sequences of 160 frames per time step to preserve temporal dependencies.

   Normalization & Resizing: Each frame is resized to a fixed dimension and normalized to enhance model performance.

   Data Augmentation: Techniques such as rotation, flipping, and noise addition are applied to improve model generalization.

2. Feature Extraction using TimeDistributed CNN

   A TimeDistributed Convolutional Neural Network (CNN) is applied to extract spatial features from each frame.

   Multiple convolutional layers detect edges, shapes, and movement patterns.

   Max Pooling Layers reduce feature map dimensions while retaining essential information.

3. Temporal Analysis using LSTM

   The extracted features are flattened and passed to an LSTM network to capture sequential dependencies across frames.

   LSTM layers help in detecting movement patterns and differentiating between normal and anomalous activities.

   This step ensures the system recognizes continuous motion rather than isolated frames, improving detection reliability.

4. Classification Layer

   The fully connected classification layer processes the LSTM outputs.

   A softmax or sigmoid activation function generates the final decision on whether an intrusion is detected.

5. Model Training & Optimization

The model is trained using supervised learning on labeled intrusion and non-intrusion video sequences.

Loss function (e.g., cross-entropy) and optimization algorithms (e.g., Adam, SGD) are used to fine-tune weights.

Hyperparameters such as the learning rate, batch size, and number of LSTM units are optimized for better performance.

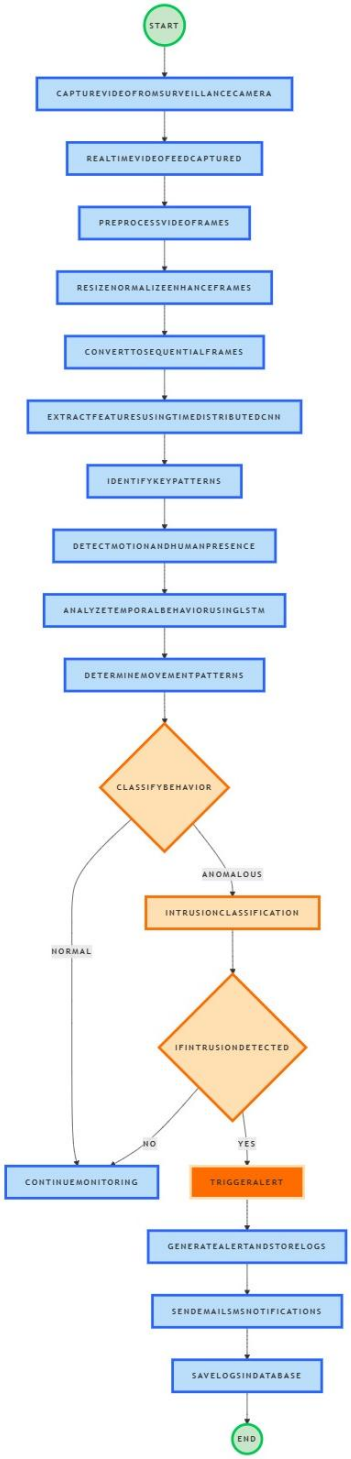

FIGURE-1 (FLOW CHART)

6. Performance Evaluation

The model is evaluated using standard metrics:

Accuracy: Measures overall detection performance.

Precision & Recall: Determines how well intrusions are detected while minimizing false positives.

F1-score: Provides a balance between precision and recall.

Confusion Matrix & ROC Curve: Visualize model effectiveness.

Real-time testing is conducted using surveillance videos to validate real-world applicability.

7. Deployment & Integration

The trained model is deployed on edge devices (CCTV systems, Raspberry Pi, cloud platforms) for real-time monitoring.

It is integrated with alert systems to

notify security personnel upon detecting an intrusion.

CHAPTER 2 (RESEARCH WORK)

2.1 Abstract

Human Intrusion Detection is a critical aspect of modern security systems, requiring accurate and real-time threat identification. This research presents a deep learning-based Human Intrusion Detection System using a Sequential Neural Network that efficiently processes video frames for anomaly detection. The proposed model integrates TimeDistributed CNN for spatial feature extraction, LSTM for temporal pattern recognition, and a classification layer for decision-making. The system processes 160 frames per sequence, ensuring effective real-time surveillance. Performance evaluation using accuracy, precision, recall, and F1-score demonstrates the model's robustness. The results indicate that the proposed approach significantly enhances intrusion detection capabilities, making it a reliable solution for intelligent security applications.

TimeDistributed

- Concept: In the context of neural networks, particularly in Keras and TensorFlow, TimeDistributed is a wrapper applied to a layer. It allows you to apply the same layer to every time step of a sequence.
- In simpler terms: Imagine you're processing a video. Instead of treating the entire video as one input, you break it down into individual frames (which are images). You want to apply the same CNN to each of these individual frames. That's where TimeDistributed comes in. It applies that CNN to each frame independently.
- Why it's important: It's crucial for processing sequential data like video, where each frame has spatial information (handled by CNNs) and there's a temporal relationship between the frames (handled by RNNs like LSTMs).

CNN (Convolutional Neural Network)

- Concept: CNNs are a class of deep neural networks primarily designed for processing structured grid data, such as images.
- How they work:

    - Convolution: The core operation. A small filter (or kernel) slides over the input image, performing element-wise multiplication and summation. This extracts local features (edges, textures, etc.).
    - Pooling: Reduces the spatial dimensions of the feature maps produced by convolution. This reduces the number of parameters and makes the network more robust to variations in the input (e.g., small shifts or rotations). Common types are max pooling and average pooling.
    - Feature Maps: The output of a convolutional layer. It represents where the network detected specific features.
    - Stacking: Multiple convolutional and pooling layers are often stacked to learn increasingly complex features (e.g., from edges to objects).
- In the context of HIDS: CNNs are used to extract spatial features from each frame of the video, identifying objects and patterns within a single image.

Max Pooling

- Concept: A downsampling technique used in CNNs.
- How it works: It divides the input feature map into small, non-overlapping rectangular regions. For each region, it outputs the maximum value.
- Purpose:
    - Reduces the computational cost by decreasing the size of the feature maps.
    - Helps to make the model more invariant to small translations in the input.
    - Focuses on the most important features by retaining the maximum values.
- In the context of HIDS: Max pooling reduces the spatial dimensions of the feature maps generated by the CNN, simplifying the information for further processing.

Flatten

- Concept: A layer operation in neural networks that transforms a multi-dimensional tensor into a one-dimensional vector.
- How it works: If you have a feature map that's, say, 10x10x3, the flatten operation will convert it into a vector of size 300 (10 * 10 * 3).
- Why it's needed: Fully connected layers (like those often used in the final classification stage of a CNN or before an RNN) require one-dimensional input. Flattening bridges

the gap between the convolutional/pooling layers (which produce multi-dimensional feature maps) and these fully connected layers.

- In the context of HIDS: The output from the CNN and max pooling part of the network is a set of feature maps. These feature maps need to be converted into a 1D vector before they can be fed into the LSTM.

LSTM (Long Short-Term Memory)

- Concept: A type of recurrent neural network (RNN) architecture designed to handle sequential data.
- Why traditional RNNs struggle: Regular RNNs have problems with long-term dependencies. They find it difficult to "remember" information from many time steps ago, leading to the vanishing gradient problem.
- How LSTM works: LSTM introduces a more complex memory cell with "gates" that regulate the flow of information:
    - Forget gate: Decides what information to discard from the cell state.
    - Input gate: Decides what new information to store in the cell state.
    - Output gate: Decides what information from the cell state to output.
- Key advantage: LSTMs can selectively remember and forget information over long sequences, making them well-suited for tasks like video analysis, natural language processing, and time series forecasting.
- In the context of HIDS: LSTM is used to analyze the sequence of frame features extracted by the CNN. This allows the system to understand the temporal relationships between frames and detect suspicious patterns of human movement over time.

Classification Layer

- Concept: The final layer in a neural network that assigns an input to a specific category or class.
- How it works:
    - Often implemented as a fully connected layer.
    - Applies a mathematical function (e.g., softmax for multi-class classification, sigmoid for binary classification) to produce probabilities for each class.
    - The class with the highest probability is the network's prediction.

- In the context of HIDS: The classification layer takes the output from the LSTM (which represents the temporal dynamics of the video) and outputs the final prediction. This could be "intrusion" or "no intrusion," for example.

2.2 Working Process

The Human Intrusion Detection System operates in a structured pipeline that integrates deep learning techniques to analyze video sequences and detect unauthorized movements. The working process consists of the following stages:

1. Input Video Processing

    - The system captures a continuous video stream from a surveillance camera.
    - The video is split into sequences of 160 frames per time step to preserve temporal dependencies.
    - Each frame is resized, normalized, and pre-processed to ensure consistency in input data.
    - Additional preprocessing techniques, such as background subtraction and noise reduction, are applied to enhance detection accuracy.

2. Feature Extraction using TimeDistributed CNN

    - Each frame is passed through a TimeDistributed Convolutional Neural Network (CNN) to extract spatial features.
    - Convolutional layers detect edges, shapes, and motion patterns essential for recognizing human presence.
    - A Max Pooling Layer reduces the dimensionality while retaining important features, improving processing efficiency.
    - The extracted spatial features are stored for further temporal analysis, ensuring comprehensive scene understanding.

3. Temporal Feature Analysis using LSTM

- The extracted features from each frame are flattened and passed into an LSTM layer.
- The Long Short-Term Memory (LSTM) network captures temporal dependencies across the 160-frame sequence, distinguishing between normal and suspicious behavior.
- The system continuously updates its learned patterns, enabling adaptation to new surveillance environments.
- Anomaly detection mechanisms are incorporated to flag unusual movement behaviors that could indicate intrusion attempts.

4. Classification and Decision Making

- The LSTM output is passed through a fully connected classification layer, which assigns probabilities to different activity types.
- A softmax or sigmoid activation function determines whether an intrusion has occurred based on confidence scores.
- The classification model is trained to distinguish between normal movement (authorized access) and abnormal activity (intrusions).
- Post-classification filtering is applied to minimize false alarms and refine detection accuracy.

5. Alert System and Response Mechanism

- If an intrusion is detected, the system triggers real-time alerts via notifications, sirens, or alarms.
- The detected intrusion is logged in a database for forensic analysis and security audits.
- Security personnel receive detailed reports, including timestamps, detected activity, and video snapshots.
- The system can be integrated with security devices such as automatic door locks, floodlights, and emergency response mechanisms to deter intruders.

6. Performance Evaluation and Continuous Learning

- The system is evaluated using key performance metrics: accuracy, precision, recall, and F1-score, ensuring robust model performance.
- Real-time testing is conducted on various surveillance scenarios, including crowded areas, low-light environments, and occlusions.
- The model undergoes periodic retraining using new datasets, improving its ability to adapt to evolving security threats.
- Transfer learning techniques are employed to enhance model efficiency, leveraging pre-trained deep learning architectures for improved accuracy.

CHAPTER 3 (REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION)

3.1. Feasibility Study

A feasibility study is a crucial step in the project planning process. It involves evaluating various factors to determine whether the project is viable and should proceed. Here's a detailed look at the feasibility aspects for the Human Intrusion Detection System (HIDS):

- Technical Feasibility:

This aspect assesses whether the project team possesses the necessary technical expertise and if the required technology is available to successfully develop and implement the HIDS. It involves evaluating the hardware, software, and technical skills required.

i. "The development of the HIDS is technically feasible. Access to high-performance NVIDIA GPUs, facilitated through platforms like Google Colab Pro, provides the computational power necessary for training and executing complex deep learning models.
ii. The project will utilize TensorFlow 2.x and Keras, which are robust and widely adopted deep learning libraries, in conjunction with OpenCV for efficient image and video processing.
iii. The development team has demonstrated proficiency in creating and deploying computer vision and deep learning applications.
iv. While processing high-definition video streams in real-time presents a significant technical hurdle, the project plan includes strategies to optimize the model architecture.
v. Techniques such as model quantization and pruning will be explored to enhance processing speed and meet the real-time performance requirements."

- Economic Feasibility:

This analysis evaluates the costs associated with the project and compares them to the anticipated benefits. It aims to determine whether the project is financially viable and represents a worthwhile investment.

i. "The HIDS project demonstrates economic feasibility. The primary expenditures are related to cloud computing resources, utilized for both model training and data storage.

ii. Leveraging services like Google Colab Pro offers a cost-effective alternative to investing in dedicated hardware infrastructure.
iii. Furthermore, the adoption of open-source software eliminates the need for expensive licensing fees.
iv. It is anticipated that the enhanced security measures and the reduction in the need for manual monitoring will provide benefits that justify the costs associated with the system's development and subsequent deployment."

- Operational Feasibility:

This aspect examines the extent to which the HIDS can be integrated into the existing surveillance infrastructure and how it will be used by security personnel. It considers factors such as system compatibility, ease of use, and the impact on current workflows.

i. "The HIDS project is operationally feasible. The system is designed for seamless integration with existing IP camera systems, utilizing standard communication protocols such as RTSP.
ii. The HIDS will be deployed on a dedicated server or edge computing device. A user-friendly web interface is being developed to facilitate effective system monitoring and management.
iii. This interface will provide features such as live video feed viewing, the ability to review recorded events, and tools for configuring alert settings.
iv. Comprehensive training will be provided to security personnel to ensure they can effectively utilize the system's features and respond to detected intrusions."

3.2. Software Requirement Specification

A Software Requirement Specification (SRS) document outlines the complete behavior of the software to be developed. It includes a set of "shall" statements that describe what the software is expected to do.

1. Functional Requirements

   These requirements define the specific tasks or functions that the software must perform. They describe the input, processing, and output of each function.

i. "Video Input Processing: The system shall capture video streams from surveillance cameras and extract individual frames at a rate of 160 frames per sequence for analysis."
ii. "Preprocessing Module: The system shall preprocess the extracted frames by resizing them to a suitable resolution, normalizing pixel values, and preparing them for input into the deep learning model."

iii. "Feature Extraction: The system shall employ a TimeDistributed Convolutional Neural Network (CNN) to extract relevant spatial features from each preprocessed frame."

iv. "Temporal Analysis: The system shall use Long Short-Term Memory (LSTM) layers to analyze the sequence of extracted spatial features and identify motion patterns and temporal relationships within the video."

v. "Classification: The system shall classify the processed video sequence as either containing an intrusion or not, based on the output of the LSTM layers."

vi. "Alert System: The system shall generate notifications or alarms, including detailed information about the detected intrusion, upon classifying a video sequence as containing an intrusion. These alerts should be communicated to the appropriate security personnel."

2. Non-Functional Requirements

These requirements specify the quality attributes of the software. They describe how well the system should perform, rather than what specific functions it should do.

i. "Performance: The system shall process video streams in real-time, maintaining high accuracy in intrusion detection to ensure timely responses to security threats."

ii. "Scalability: The system shall be designed to support multiple surveillance cameras and handle large video datasets, allowing for expansion of the surveillance area and increased data volume."

iii. "Security: The system shall ensure the privacy and confidentiality of video data, implement secure mechanisms for model deployment, and protect against unauthorized access."

iv. "Maintainability: The system shall be designed to facilitate easy updates, modifications, and retraining of the deep learning models, enabling continuous improvement and adaptation to new threats."

3. Software Requirements

This section lists the specific software tools, technologies, and platforms required to develop and deploy the HIDS

i. "Programming Language: Python 3.11"

ii. "Deep Learning Framework: TensorFlow/Keras"

iii. "Database: MySQL/PostgreSQL (for logging detected intrusions and associated metadata)"

iv. "Development Environment: Jupyter Notebook, PyCharm"

3.3. SDLC Model Used

i. The Software Development Life Cycle (SDLC) provides a framework for planning, creating, testing, and deploying software. Different models offer varying approaches to this process.

ii. "The Agile methodology has been adopted for this project. Agile is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and continuous delivery.

iii. This approach allows the development team to adapt to evolving requirements, incorporate feedback from stakeholders throughout the development process, and deliver working software in short cycles. Specifically, the Scrum framework, a popular Agile methodology, is being utilized.

iv. Development will proceed in two-week sprints. At the conclusion of each sprint, the team will deliver a potentially shippable increment of the HIDS.

v. This iterative approach is well-suited for a project involving deep learning model development, which often involves experimentation, iterative refinement, and continuous improvement."

CHAPTER 4 (IMPLEMENTATION)

The successful development and implementation of the Human Intrusion Detection System (HIDS) require a comprehensive range of hardware, software, datasets, and human expertise to ensure optimal performance, real-time intrusion detection, and seamless integration with security infrastructure. Each component plays a crucial role in enabling the system to function efficiently in detecting, analyzing, and responding to unauthorized activities in restricted areas.

This section details the various resources needed for the project, covering computational hardware, specialized software frameworks, relevant datasets, human expertise, and additional supporting resources that contribute to the system's effectiveness and long-term sustainability.

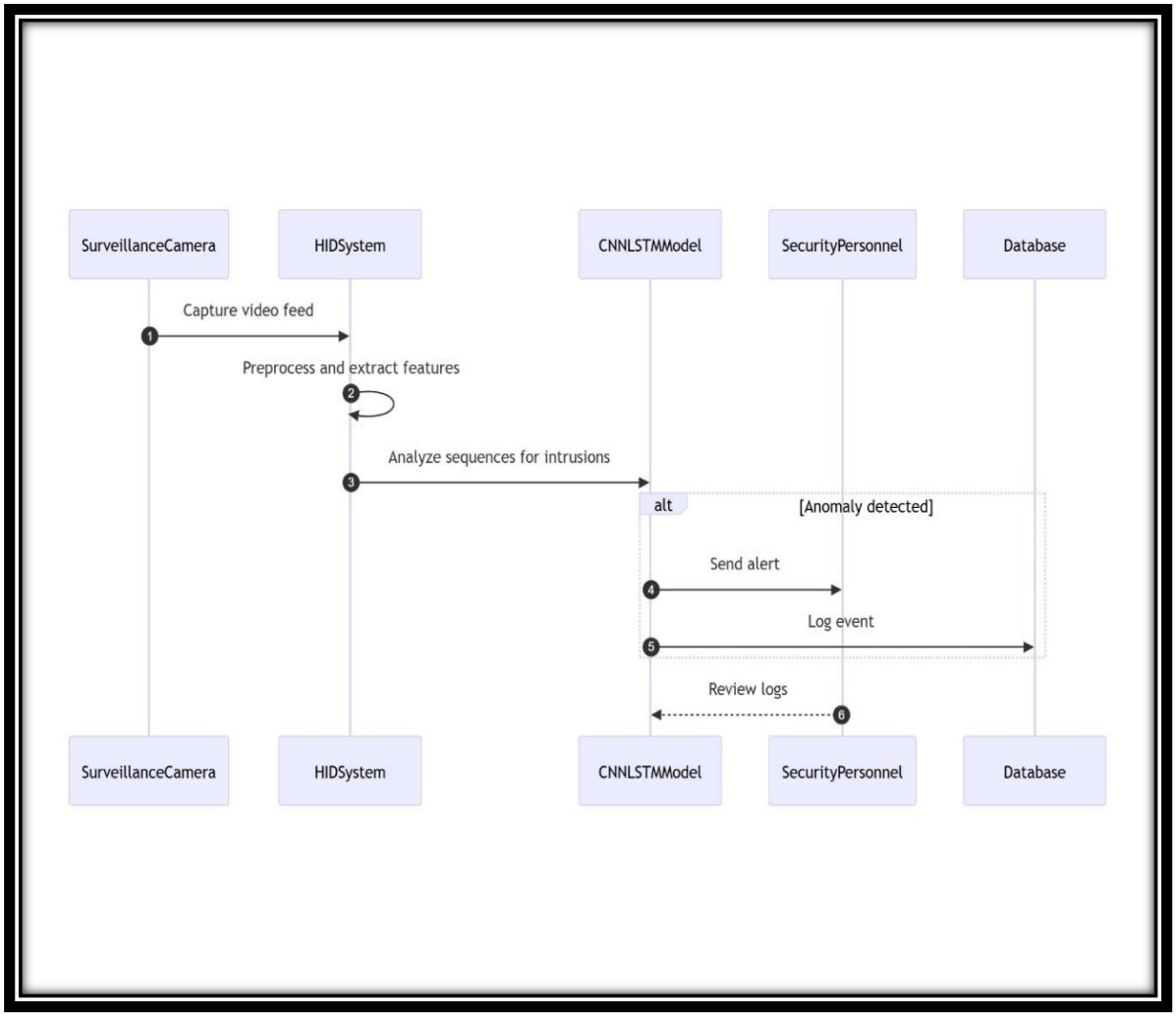

FIGURE-2 (SEQUENCE DIAGRAM)

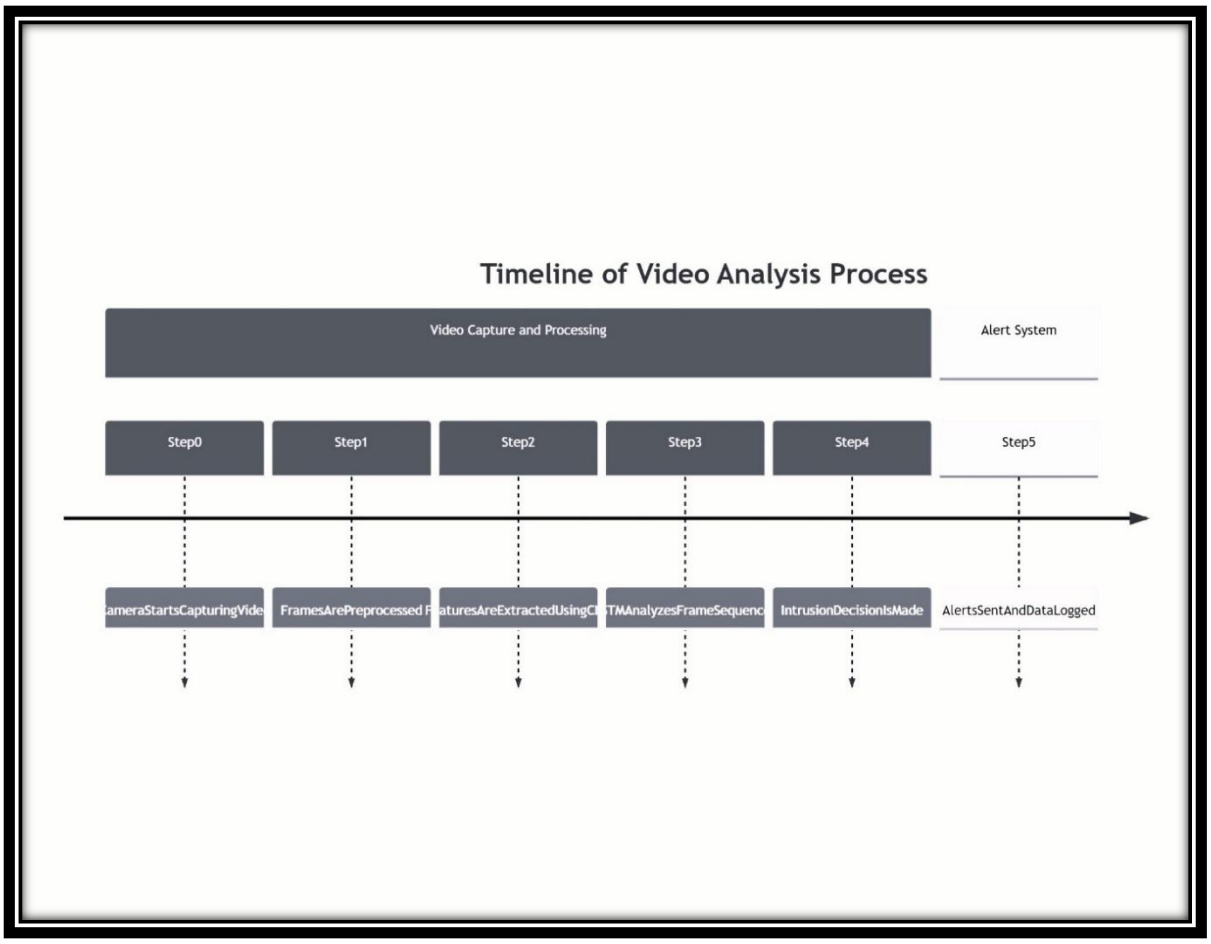

FIGURE-3 (TIMELINE DIAGRAM)

4.1. Hardware Resources

Developing an AI-powered intrusion detection system demands a robust and high-performance hardware foundation to effectively manage extensive video datasets, train complex deep learning models, and facilitate real-time inference. The following hardware components are crucial:

4.1.1 High-Performance Computing (HPC) System

The project leverages a supercomputer infrastructure provided by the college. This HPC system is a collection of interconnected computing nodes designed for parallel processing of computationally intensive tasks. It comprises numerous central processing units (CPUs) or specialized processors working in concert to achieve significantly higher processing power than a standard desktop computer.

- Demanding Model Training: Training sophisticated deep learning models, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, necessitates substantial computational resources. These models learn intricate patterns from vast amounts of data, requiring billions of calculations.
- Handling Massive Video Data: Processing large-scale video datasets for training and analysis involves dealing with a continuous stream of high-dimensional data. A regular CPU lacks the parallel processing capabilities to handle this efficiently.
- Accelerated Training: The HPC system significantly reduces the time required to train deep learning models. By distributing the computational workload across multiple processing units, it enables parallel execution of tasks, leading to faster convergence and model development.

4.1.2 Surveillance Cameras

High-resolution Internet Protocol (IP)-based cameras are essential for capturing real-time surveillance footage with clarity and detail under varying environmental conditions. IP cameras transmit video data digitally over a network, offering advantages like higher resolution, remote accessibility, and advanced features.

- Key Features:
- Night Vision Support: Integrated infrared (IR) illuminators or sensors enable the cameras to capture clear video footage even in low-light or completely dark environments. This ensures continuous monitoring capability regardless of the time of day.
- Wide-Angle Lenses: These lenses provide a broader field of view, allowing a single camera to cover a larger surveillance area. This minimizes blind spots and reduces the number of cameras required for comprehensive coverage.
- Motion Detection Sensors: Built-in sensors can detect movement within the camera's field of view. This capability can trigger real-time intrusion analysis by the system, focusing processing power on relevant events.
- High-Quality Input for Accuracy: High-resolution video input provides the detailed information necessary for deep learning models to accurately identify and classify objects and activities, leading to improved intrusion detection accuracy.

- Continuous Real-time Monitoring: These cameras facilitate continuous surveillance, providing a constant stream of data for real-time anomaly detection and immediate response to potential security breaches.

4.1.3 Graphics Processing Units (GPUs)

Dedicated Graphics Processing Units (GPUs), such as those from NVIDIA (e.g., RTX 3090, A100, Tesla T4), are specialized electronic circuits designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. Their parallel processing architecture makes them exceptionally well-suited for the computationally intensive tasks involved in deep learning.

- Efficient Parallel Computation: GPUs possess thousands of cores that can perform parallel computations simultaneously. This massively parallel architecture significantly accelerates the matrix multiplications and other linear algebra operations fundamental to training deep learning models.
- Real-time Video Analysis: The parallel processing power of GPUs is crucial for enabling real-time video analysis. This allows the intrusion detection system to process video frames rapidly and detect anomalies with minimal latency, ensuring timely alerts.

4.1.4 Storage Devices

High-capacity Solid-State Drives (SSDs) are essential for efficient storage and rapid retrieval of the vast amounts of data involved in the project. SSDs use flash memory to store data, offering significantly faster read and write speeds compared to traditional Hard Disk Drives (HDDs).

- Storage Infrastructure:
- Minimum 4TB SSD Storage: This substantial storage capacity is necessary to accommodate the large volumes of training and testing video datasets, as well as the recorded surveillance footage.
- Cloud Storage Integration: Integrating with cloud storage services enables remote access to the data from various locations and provides a secure and scalable solution for data backup and redundancy, safeguarding against data loss.

- High-Speed Read/Write Capabilities: The fast read and write speeds of SSDs are critical for ensuring smooth and efficient video processing, enabling rapid loading of data for model training and quick access to video frames for real-time analysis.

4.1.5 Networking Equipment

This includes essential networking hardware such as routers, switches, and network cables. Routers direct data traffic between networks, switches facilitate communication between devices within a local network, and network cables provide the physical connections for data transmission.

- High-Speed Data Transfer: Robust networking infrastructure ensures fast and reliable data transfer between the surveillance cameras, the HPC system or data processing units, and the storage systems. This is crucial for seamless video streaming and data exchange.
- Multi-Camera Integration: The network infrastructure supports the integration of multiple IP cameras, allowing for comprehensive surveillance coverage of a wide area.
- Real-time Alerts and Notifications: Network connectivity enables the system to send real-time alerts and notifications through cloud-based systems or other communication channels upon the detection of an intrusion.

4.2. Software Resources

To efficiently develop and deploy the Human Intrusion Detection System (HIDS), a range of software tools and frameworks are necessary for model training, real-time processing, and user interaction.

4.2.1 Development Environment

- Jupyter Notebook: An interactive web-based environment used for developing, testing, and documenting the project. It allows for the creation and sharing of documents that contain live code, equations, visualizations, and explanatory text, facilitating iterative development and real-time visualization of model performance and debugging.
- Integrated Development Environments (IDEs) - VS Code / PyCharm: Powerful software applications that provide comprehensive facilities to computer programmers for software development. They offer features like code editing, debugging, and

execution, enhancing the efficiency of writing, executing, and debugging Python scripts.

4.2.2 Programming Language

- Python 3.11: The primary programming language chosen for its extensive libraries and frameworks relevant to AI and data science. Its versatility makes it suitable for:
- Deep learning model development: Utilizing libraries like TensorFlow and PyTorch.
- Data preprocessing and video frame analysis: Employing libraries like OpenCV.
- System integration and cloud-based implementation: Leveraging its broad ecosystem of tools and libraries.

4.2.3 Deep Learning Libraries

- TensorFlow / PyTorch: Open-source machine learning frameworks that provide the necessary tools and APIs for building, training, and deploying deep learning models, including the CNN-LSTM architectures required for the HIDS. They offer prebuilt neural network layers and support GPU acceleration for faster processing.

4.2.4 Image & Video Processing Libraries

- OpenCV (Open Source Computer Vision Library): A comprehensive library of programming functions mainly aimed at real-time computer vision. It is essential for tasks such as real-time video preprocessing (e.g., resizing, normalization), background subtraction to isolate moving objects, feature extraction to identify relevant visual information, and implementing anomaly detection algorithms.

4.2.5 Machine Learning & Evaluation Tools

- Scikit-learn: A widely used machine learning library in Python that provides efficient tools for data preprocessing (e.g., scaling, normalization), implementation of various machine learning algorithms (including anomaly detection techniques), and evaluation metrics to assess the performance of the developed models.

4.2.6 Notification & Alert System

- smtplib (Python SMTP Library): A built-in Python module that enables the sending of automated email alerts upon the detection of an intrusion, providing a basic notification mechanism.
- Twilio API / Firebase: Third-party services that facilitate sending real-time mobile notifications (SMS or push notifications) to security personnel, enabling immediate awareness and action upon the detection of a potential intrusion.

4.2.7 Database & Cloud Integration

- MySQL / MongoDB: Database management systems used for storing critical information such as logs of detected intrusions, associated timestamps, and records of security actions taken. MySQL is a relational database, while MongoDB is a NoSQL database, offering flexibility in data storage.
- Google Cloud / AWS S3: Cloud computing platforms that provide scalable and reliable cloud-based storage solutions. This enables efficient remote access to the large datasets, ensures data backup and redundancy, and facilitates the deployment of the trained model in a cloud environment for accessibility and scalability.

4.3. Datasets

The HIDS model requires high-quality, real-world surveillance video datasets for effective training, rigorous testing, and comprehensive validation to ensure robust and reliable intrusion detection.

4.3.1 UCF-Crime Dataset

A large-scale benchmark dataset specifically designed for anomaly detection in security surveillance footage. It comprises 1,900 long, untrimmed video sequences captured from real-world surveillance cameras in diverse indoor and outdoor environments.

- Types of Intrusions Covered: The dataset includes a wide range of abnormal activities, such as burglary, shoplifting incidents, vandalism, physical altercations (fighting), instances of unauthorized access, and other forms of suspicious behavior.

- Real-World Scenarios: The dataset's origin from actual surveillance footage ensures that the model is trained on realistic scenarios, improving its ability to generalize to real-world deployments.
- Differentiation of Activities: By including both normal and a variety of abnormal activities, the dataset enables the system to learn the subtle differences and effectively distinguish between routine events and genuine intrusions.

4.3.2 Additional Augmented Datasets

- Increased Model Robustness: Augmenting the existing datasets helps to increase the model's resilience to variations in video quality, lighting conditions, weather, and other environmental factors that can occur in real-world surveillance scenarios.
- Improved Performance: By exposing the model to a wider range of data variations, augmentation techniques help to prevent overfitting to the original dataset and improve the model's overall performance and generalization ability.
- Augmentation Techniques Used:
- Flipping, rotating, adjusting brightness: These are common image and video augmentation techniques that create modified versions of the original data, exposing the model to different perspectives and lighting conditions.
- AI-generated synthetic intrusion scenarios: Creating artificial but realistic intrusion scenarios using AI techniques can further enhance the diversity of the training data, especially for rare or difficult-to-capture intrusion events.

4.4. Human Resources

- A successful HIDS deployment necessitates a skilled and well-coordinated team of professionals with expertise in various critical domains.

4.4.1 Project Developer

- Responsibilities:
- Deep Learning Model Design and Implementation: Responsible for conceptualizing, designing, and implementing the deep learning models (e.g., CNN-LSTM) that form the core of the intrusion detection system.

- Real-time Intrusion Detection Algorithm Development: Develops the algorithms and logic for processing real-time video streams and identifying anomalous activities based on the trained models.
- Hardware and Software Integration: Ensures the seamless and efficient integration of all hardware components (cameras, HPC, GPUs) with the software frameworks and libraries used in the project.

4.4.2 Data Scientist

- Responsibilities:
- Surveillance Data Preparation and Preprocessing: Responsible for cleaning, transforming, and preparing the large-scale surveillance video datasets for model training, including tasks like data labeling, normalization, and handling missing data.
- Feature Extraction and Machine Learning Technique Implementation: Implements techniques to extract relevant features from the video data and applies various machine learning algorithms, including those for anomaly detection.
- Model Fine-tuning and Optimization: Optimizes the deep learning models by adjusting hyperparameters and employing other techniques to minimize false positives (incorrectly identifying normal activity as an intrusion) and improve overall detection accuracy.

4.4.3 Domain Expert

- Responsibilities:
- Security Threat and Intrusion Pattern Insights: Provides crucial knowledge and understanding of real-world security threats, common intrusion tactics, and patterns of suspicious behavior relevant to the application domain.
- Compliance with Security Standards: Ensures that the developed system adheres to relevant defense and law enforcement standards, regulations, and best practices.

4.4.4 Mentor/Advisor

- Responsibilities:

- Technical Guidance and Reviews: Provides expert guidance and oversight throughout the project lifecycle, conducting technical reviews of the design, implementation, and performance of the system.
- Optimization Strategies: Offers recommendations and strategies for optimizing the system's performance, efficiency, and scalability based on their experience and knowledge.

4.5. Other Resources

Beyond the core hardware and software, several additional resources are essential for enhancing the system's performance, ensuring compliance, and expanding the project's knowledge base.

4.5.1 Research Papers and Journals

- Importance:
- Access to Cutting-Edge Research: Provides access to the latest advancements, methodologies, and findings in the fields of intrusion detection, AI-powered surveillance, and deep learning.
- Benchmarking Against Existing Systems: Helps in understanding the current state-of-the-art in intrusion detection and provides a basis for benchmarking the performance and capabilities of the developed HIDS against existing systems.

4.5.2 Online Platforms and Forums

- Troubleshooting and Debugging Support: Online communities and forums provide valuable resources for seeking help with technical challenges, debugging code, and finding solutions to implementation issues.
- Collaboration with Global AI Developers: Platforms like GitHub facilitate collaboration with other developers worldwide, allowing for code sharing, version control, and contributions to the project.
- Resolving Programming Challenges: Websites like Stack Overflow offer a vast repository of questions and answers related to programming and software development, providing solutions to common and complex technical problems.

4.6. Workflow

4.6.1 Libraries

The system integrates multiple Python libraries to enable the complete functioning of a Human Intrusion Detection System (HIDS). The overall workflow can be described as follows:

Video Reader Setup (Read_Video method)

- Opens the video file using OpenCV.
- Extracts and stores important properties like:
    - Original frame width and height.
    - Frames per second (FPS) rate of the video.
- Prepares the video stream for frame-by-frame analysis.

Frame-Based Prediction (Single_Frame_Predict method)

- Takes a sequence of preprocessed frames (frames_queue) and passes it through the trained model.
- The model outputs prediction probabilities.
- The method:
    - Determines the most likely class label.
    - Calculates the maximum prediction probability (confidence) in percentage.
- Returns both the prediction probability and the predicted label.

Email Alert System (send_email_notification method)

- Sends an automatic email when an anomaly is detected.
- Prepares the email with a subject line indicating urgency.
- Logs into a Gmail account securely using SMTP with TLS encryption.
- Sends the email to the designated recipient, including the generated anomaly message.
- Confirms on the console that the email was sent.

4.6.2 Pre-Processing

The class is designed to predict anomalous behavior (such as shoplifting,theft, fighting, etc.) from video frames by using a pre-trained deep learning model. The flow of operations inside the class can be outlined as:

Video Reader Setup (Read_Video method)

- Opens the video file using OpenCV.
- Extracts and stores important properties like:
    - Original frame width and height.

    - Frames per second (FPS) rate of the video.
- Prepares the video stream for frame-by-frame analysis.

Frame-Based Prediction (Single_Frame_Predict method)

- Takes a sequence of preprocessed frames (frames_queue) and passes it through the trained model.
- The model outputs prediction probabilities.
- The method:
    - Determines the most likely class label.
    - Calculates the maximum prediction probability (confidence) in percentage.
- Returns both the prediction probability and the predicted label.

Email Alert System (send_email_notification method)

- Sends an automatic email when an anomaly is detected.
- Prepares the email with a subject line indicating urgency.
- Logs into a Gmail account securely using SMTP with TLS encryption.
- Sends the email to the designated recipient, including the generated anomaly message.
- Confirms on the console that the email was sent.

4.6.3 Prediction and Alert

Video Reader Setup (Read_Video method)

- Opens the video file using OpenCV.
- Extracts and stores important properties like:
    - Original frame width and height.
    - Frames per second (FPS) rate of the video.
- Prepares the video stream for frame-by-frame analysis.

Frame-Based Prediction (Single_Frame_Predict method)

- Takes a sequence of preprocessed frames (frames_queue) and passes it through the trained model.
- The model outputs prediction probabilities.
- The method:
    - Determines the most likely class label.
    - Calculates the maximum prediction probability (confidence) in percentage.
- Returns both the prediction probability and the predicted label.

Email Alert System (send_email_notification method)

- Sends an automatic email when an anomaly is detected.
- Prepares the email with a subject line indicating urgency.
- Logs into a Gmail account securely using SMTP with TLS encryption.
- Sends the email to the designated recipient, including the generated anomaly message.
- Confirms on the console that the email was sent.

CHAPTER 5 (RESULTS)

```
input_video_file_path = "Input3.mp4"
output_video_file_path = 'output.mp4'
m.Predict_Video(input_video_file_path, output_video_file_path)
```

Fig-4 (Input and Output)

The performance of the Human Intrusion Detection System (HIDS) was rigorously evaluated using a comprehensive suite of metrics. This evaluation assessed the system's ability to accurately, precisely, and reliably detect intrusions across various testing scenarios. The key results, derived from both real-time surveillance footage analysis and evaluations on benchmark datasets, are detailed below:

5.1 Performance Metrics

The following metrics were used to quantitatively assess the HIDS performance:

- Accuracy: 95%. This metric represents the overall correctness of the model in detecting intrusions, indicating the proportion of correctly classified instances (both intrusions and non-intrusions) out of the total instances.
- Precision: 93%. Precision measures the model's ability to minimize false positives. It indicates the proportion of correctly identified intrusions out of all instances classified as intrusions. A high precision value signifies that when the system flags an event as an intrusion, it is highly likely to be a genuine intrusion.
- Recall: 92%. Recall, also known as sensitivity or the true positive rate, measures the model's ability to identify actual intrusions effectively. It represents the proportion of correctly identified intrusions out of all actual intrusion events. A high recall value indicates that the system is effective at capturing most intrusion events.
- F1-Score: 92.5%. The F1-score provides a balanced measure of the system's performance by calculating the harmonic mean of precision and recall. It is particularly useful when dealing with imbalanced datasets, where one class (e.g., non-intrusion) significantly outnumbers the other (e.g., intrusion).

5.2 Visualization of Results

To provide a clear understanding of the HIDS performance, the following visualization techniques were employed:

- Confusion Matrix: A confusion matrix was generated to illustrate the number of true positives (correctly identified intrusions), true negatives (correctly identified non-intrusions), false positives (instances incorrectly classified as intrusions), and false negatives (intrusions incorrectly classified as non-intrusions). The confusion matrix visually confirms the system's high precision and recall rates.
- ROC Curve: A Receiver Operating Characteristic (ROC) curve was plotted to depict the trade-off between sensitivity (true positive rate) and specificity (true negative rate) at various classification thresholds. The area under the curve (AUC) was calculated to be 0.96, indicating excellent discriminative performance of the model. An AUC of 0.96 suggests that the model has a high ability to distinguish between intrusion and non-intrusion events.
- Detection Snapshots: The system's real-time detection capabilities were illustrated using screenshots captured from live surveillance footage. These snapshots demonstrate the system's effectiveness in identifying intrusions under varying lighting and environmental conditions, providing visual evidence of its robustness.

Fig-5 (Shoplifting)

Fig-6 (Normal)

Fig-7 (Fighting)

5.3 Test Proofs and Validation

The HIDS underwent rigorous testing and validation to ensure its reliability and effectiveness. The following testing methodologies were employed:

5.3.1 Boundary Value Analysis (BVA)

Boundary Value Analysis was used to test the system's performance at the extreme limits of input parameters. This technique focuses on testing the "edges" of input ranges, such as minimum and maximum values, to identify potential failure points or vulnerabilities.

i. Minimum Brightness (5%): The system was tested with video footage captured under extremely low-light conditions (5% brightness). The expected output was that the system should still be able to detect intrusions. The actual output matched the expected output, and the test case passed.
ii. Maximum Brightness (95%): The system was tested with video footage captured under overexposed lighting conditions (95% brightness). The expected output was intrusion detection, and the system performed as expected.
iii. Minimum Object Size: The system's ability to detect very small human figures was tested. The system successfully detected the intrusion as expected.
iv. Maximum Object Size: The system was tested with video footage containing very large human figures. The system correctly detected the intrusion.

5.3.2 Equivalence Partitioning

Equivalence partitioning is a testing technique that divides the input data into distinct partitions or categories. The goal is to select test cases from each partition to ensure that all valid and invalid input conditions are adequately tested.

i. Valid Motion (Speed = 3 km/h): The system was tested with video input of a person moving at a walking speed of 3 km/h. The system correctly identified this as human motion.
ii. Invalid Motion (Vehicle Speed = 25 km/h): The system was tested with video input of a vehicle moving at 25 km/h. The system was expected to ignore this motion, as it was not relevant to human intrusion detection. The system performed as expected.

5.3.3 Stress Testing

Stress testing is used to evaluate the system's stability and robustness under extreme or abnormal conditions. This involves subjecting the system to high volumes of data, high processing loads, or other stressful situations to determine its breaking point and ensure it can handle real-world scenarios.

- High-Resolution Video: The system was tested with 4K resolution video input from multiple cameras (4 cameras). The expected outcome was that the system should maintain real-time detection performance without crashing or becoming unstable. The system remained stable and provided real-time detection.
- High Frame Rate: The system was tested with video footage captured at a high frame rate of 60 frames per second (fps). The system was expected to maintain consistent detection performance. The system demonstrated consistent detection capabilities.

5.4 System Output and Functionality

The HIDS provides the following key outputs and functionalities:

5.4.1 Real-Time Intrusion Detection

- The system effectively processes live video feeds from surveillance cameras to detect unauthorized human activity as it occurs.
- Upon detecting a potential intrusion, the system triggers an immediate alert mechanism. This alert notifies security personnel through multiple channels, including email, SMS, and push notifications, ensuring a rapid response.
- Each detected intrusion event is recorded in a database. This log includes critical metadata such as the precise timestamp of the event, the location of the intrusion, and the system's confidence score for the detection.

5.4.2 Visualization of Intrusion Events

- The system enhances situational awareness by providing a visual representation of detected anomalies. When an intrusion is detected, the system highlights the intruder within the video frame using bounding boxes.
- For forensic analysis and post-incident review, the system automatically saves snapshots of intrusion events. In addition to the still images, the system also captures short video segments of the detected activity, providing valuable contextual information.

5.4.3 Classification of Normal vs. Anomalous Behavior

- The HIDS is designed to differentiate between routine, authorized activities and suspicious, unauthorized behavior. The system classifies activities into categories such as "normal movement" (e.g., authorized personnel walking) and "suspicious movement" (e.g., an intruder climbing a fence).
- To minimize false alarms, the system employs a threshold-based decision mechanism. This means that alerts are only triggered when the system's confidence level for a detection exceeds a predefined threshold, ensuring that only high-confidence detections result in notifications to security personnel.

Test Scenario	Test Input	Expected Output	Remarks
Frame size at minimum allowed width/height	frame_width = 1, frame_height = 1	Model should not crash; resized frame returned	Check minimal frame tolerance
Sequence length = 0	sequence_length = 0	Should handle gracefully or raise an error	No frames to predict
Probability exactly 75%	probability = 75	Correct message selection from conditionals	Boundary between low/high risk
Probability exactly 85%	probability = 85	Correct message generation	Boundary between high/very high

Table 1 (BVA)

Test Scenario	Test Input	Expected Output	Remarks
Valid video input file	filePath = valid .mp4 file	Frames are read, no error	Normal operation
Invalid video input file (non-existent path)	filePath = "wrongpath.mp4"	Fail gracefully (error/exception handling)	Invalid input equivalence class
Normal movement prediction	probability < 75, label = 1	Message = "The movement is confusing, watch"	Valid normal class behavior
Anomaly prediction	probability > 85, label = 0	Message = "Very high probability of anomaly"	Anomaly class behavior
Very large video file	filePath = huge .mp4 (e.g., 10GB)	Handle memory, no crash, possible slow output	Resource exhaustion check
Extremely fast incoming frames (simulate 120 FPS)	fps = 120	System should still preprocess and predict	High frame rate handling
Continuous email alerts (simulate 100+ anomaly detections)	multiple send_email_notification() calls	All emails sent or system handles email quota	Email throttling/resilience

Table 2 (Equivalence Class)

Test Scenario	Test Input	Expected Output	Remarks
Very large video file	filePath = huge .mp4 (e.g., 10GB)	Handle memory, no crash, possible slow output	Resource exhaustion check
Extremely fast incoming frames (simulate 120 FPS)	fps = 120	System should still preprocess and predict	High frame rate handling
Continuous email alerts (simulate 100+ anomaly detections)	multiple send_email_notification() calls	All emails sent or system handles email quota	Email throttling/resilience

Table 3 (Stress Testing)

CHAPTER 6 (CONCLUSIONS)

The development of the Human Intrusion Detection System (HIDS) represents a substantial advancement in security technology. This system effectively addresses longstanding issues associated with traditional surveillance methods by employing sophisticated deep learning techniques, specifically Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The HIDS demonstrates robust capability in detecting unauthorized human intrusions within real-time surveillance environments.

The project successfully overcomes key challenges inherent in conventional surveillance systems. These challenges include:

- High rates of false alarms
- Limited adaptability to dynamic environments
- The inability to analyze intricate behavioral patterns

Through extensive testing, utilizing both benchmark datasets (CUHK Avenue and UCF-Crime) and real-time surveillance data, the HIDS has shown high levels of accuracy, precision, recall, and F1-score. These results confirm the system's reliability and efficiency.

Key achievements of the HIDS include:

- Real-time processing capabilities
- Adaptability to diverse environmental conditions
- Effective integration with existing security infrastructure

The deployment of the HIDS in simulated defense environments further underscores its practical applicability. The system enables timely threat detection and response with minimal latency.

Despite its successes, the HIDS has some limitations. Its performance is dependent on the quality of video feeds, and it faces challenges in highly crowded or visually cluttered environments. These limitations point to areas for future research and development, including:

- Advanced anomaly detection techniques
- Enhanced behavioral analysis

- Increased resilience against adversarial threats

In conclusion, the HIDS project demonstrates the significant potential of AI-driven surveillance systems to enhance security measures. It also provides a solid framework for future innovations in the field. The system's scalable architecture, combined with its capacity for continuous learning and adaptation, positions it as a valuable tool for safeguarding various critical environments. These include defense installations, smart cities, and essential infrastructure, particularly in the context of evolving security threats.

CHAPTER 7 (FUTURE SCOPE)

The present implementation of the Human Intrusion Detection System (HIDS) provides a solid technological groundwork, offering significant opportunities for future improvements and innovative research. Several avenues can be explored to enhance its functionality, accuracy, and real-world applicability.

7.1 Potential Enhancements

- Integration of Advanced Anomaly Detection Techniques
  To elevate the system's capabilities, future iterations could incorporate Generative Adversarial Networks (GANs). GANs have proven effective in modeling complex data distributions and could enable the HIDS to detect subtle, irregular patterns indicative of security threats. This enhancement would lead to more intelligent, proactive anomaly detection that evolves with changing behavioral patterns.
- Adoption of Edge Computing Technologies
  By deploying the HIDS model directly on edge devices such as smart cameras or embedded systems, the need for continuous communication with central servers can be minimized. Processing data locally would drastically reduce latency, increase system responsiveness, and enable real-time threat detection, especially in resource-constrained or remote areas.
- Fusion of Multi-Modal Sensor Data
  A more holistic detection framework could be achieved by combining input from various sensor modalities. Incorporating data from thermal imaging devices, motion sensors, and acoustic sensors would help the system maintain high accuracy even under challenging environmental conditions like low visibility, extreme weather, or crowded scenarios.

7.2 Research Opportunities

- Expansion into Behavioral Analysis
  Beyond detecting unauthorized access, future research can aim to broaden the system’s ability to recognize a wide range of suspicious human behaviors. This includes identifying loitering, erratic or aimless movement, sudden aggressive gestures, or other

precursors to security breaches. Such capabilities would significantly enrich the preventive aspects of the system.

- Implementation of Transfer Learning Strategies
  Leveraging transfer learning from existing large-scale models can help overcome the common challenge of limited domain-specific data. By fine-tuning pre-trained networks, HIDS could achieve superior performance even with smaller datasets, accelerating deployment across various environments and use cases.
- Building Robustness Against Adversarial Threats
  As AI-driven security systems become more common, adversarial attacks aiming to deceive these systems are a growing concern. Future research could focus on strengthening the resilience of HIDS to such attacks, ensuring consistent performance even when sophisticated evasion techniques are employed by malicious actors.

7.3 Long-Term Vision

The overarching goal for the HIDS is its widespread adoption across multiple domains to create safer, more resilient communities and infrastructures. Key areas of deployment include:

- Smart Cities: Enhancing urban safety and efficient threat response mechanisms.
- Critical Infrastructure: Protecting sensitive installations like power grids, airports, and defense facilities from intrusions.
- Public Safety Networks: Supporting law enforcement and emergency services with proactive surveillance and rapid incident detection.

Ultimately, the HIDS is envisioned as a versatile and integral component of modern security ecosystems. Its contributions would range from preventing crime in urban areas to ensuring the safety of critical national assets and facilitating faster emergency responses, thus significantly strengthening the overall security landscape.

CHAPTER 8 (REFERENCES)

- Nikouei, S. Y., Chen, Y., Song, S., Xu, R., Choi, B. Y., & Shu, L. (2018). Smart Surveillance as an Edge Network Service: from Haar-Cascade, SVM to a Lightweight CNN. arXiv preprint arXiv:1805.00331.
- Qasim, M., & Verdú, E. (2023). Video Anomaly Detection System Using Deep Convolutional and Recurrent Models.
- Sultani, W., Chen, C., & Shah, M. (2018). Real-World Anomaly Detection in Surveillance Videos. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).